RESEARCH ARTICLE OPEN ACCESS

Structural Optimization Using Multi-Objective Genetic Algorithm

Jonathas I. F. Oliveira*, Antonio C. O. Miranda**
**(Building Construction Technology, Federal Institute of Ceará, Juazeiro do Norte-CE, Brazil*
*** (College of Technology, University of Brasília, Brasília-DF, Brazil*
Correspondingf author : Jonathas I. F. Oliveira

ABSTRACT
Structural design is a process in which the designer must decide the topology of a structural system, materials to be used, arrangement of the elements and its respective dimensions. In essence, it is an iterative process. It begins with a choice made by the designer, based on his experience and technical recommendations, and is continually altered until certain project criteria are achieved, such as maximum allowable displacement and stresses. In reinforced concrete structures, the positioning of the columns is one of the choices that the designer must make. This paper implements optimization techniques, such as genetic algorithms, to find the optimum column placement in reinforced concrete frames and to provide design alternatives. By altering the dimensions of the elements and the positions of the columns, a multi-objective genetic algorithm is used to minimize costs associated with concrete, steel reinforcements, shapes and maximum displacement. This paper deviates from others present in the literature regarding the way the columns are positioned and its applications in non-regular topological structures.
Keywords – Column Positioning, Multi-objective optimization, structures.

Date of Submission: 25-02-2020 Date Of Acceptance: 05-03-2020

I. INTRODUCTION

The design of conventional reinforced concrete structures can be simplified into four iterative stages: conception, analysis, sizing and detailing. Although this is a sequential procedure, the process as a whole is iterative and the steps defined herein are interdependent.

Currently, the design of conventional reinforced concrete structures is supported by specialized computer programs, which allow the design of complex models and provide superior productivity, offering the designer greater freedom in the design and study of the structure behavior. These programs now automate a large part of the project preparation process, operating mainly at the stages of analysis, element sizing and detailing. However, the conception phase and structural design phase still have a high level of dependency on the designer. The available programs still require the user to manually determine the position of the elements, their geometric characteristics, materials, element connectivity and other fundamental characteristics for the definition of the structure model.

Optimization techniques have been implemented in several problems in the conception and design of structures. Although these are techniques employed in the search for optimal solutions, many of them are not necessarily implemented with this purpose in mind. Optimization can also be employed in the search for alternative solutions that may be more attractive than well-known solutions, and that is the reason for its use in this paper.

The purpose of this research is to apply a multi-objective optimization technique to the design of structures, especially reinforced concrete frames structures. It is intended to disclose a methodology for the exploration of the project solution space, in order to present and propose alternative design solutions, aiming at a better performance of the structure. The search for these solutions will take place by pursuing a better positioning of the columns and variation of the transversal sections of the elements, considering that the positioning of the beams of a floor are informed before the optimization process.

II. HEADINGS

Optimization techniques are widely applied in engineering problems. For example, [1] used the conjugate gradient method for optimizing aircraft wing shapes. [2] studied the optimization of aircraft engine suspension systems seeking a better vibration insulation and reduction of excessive displacements. [3] studied the optimization of turbine positioning in wind farms, seeking greater energy production, considering the uncertainties. Optimization

techniques were also employed in conjunction with neural networks in the work of [4], aiding in the detection of travel destinations from travel surveys.
The papers above show how versatile optimization techniques can be, given that these are applied in several knowledge fields for problem solving.

2.1. STRUCTURAL OPTIMIZATION

Unlike steel structures, reinforced concrete structures present a greater variety and complexity in the characteristics of the material, which directly influences its behavior and analysis. Due to that, the process of optimizing concrete structures tends to be more complex than those involving steel structures.
Among the variables considered in the problems of concrete structure are: cross-sectional dimensions, concrete resistance to compression, characteristic yield resistance of steel, amount of reinforcement in the sections, shape and topology of the structure.

Most of the work involving optimization of concrete structures uses the approach of minimizing the structure's cost, as can be seen in [5]. In these situations, the cost of the structure is composed by the costs related to the amount of concrete, steel and formwork. The concrete's cost is calculated according to the volume of concrete required; that of steel depends on its weight and that of shapes is dependent of its total area, multiplied by their respective unit costs. It is important to emphasize that labor costs, to simplify, will be embedded in unit prices, as done by [6].

The constraints on concrete structure optimization problems include: minimum and maximum amount of reinforcement in a cross section, limiting displacements, checking the bearing capacity, minimum and maximum section's dimensions, maximum quantity of different sections, among others.

In the last decades, meta-heuristic algorithms, genetic algorithms among them, have been widely used in optimization problems. These algorithms seek to mimic the genetic mechanisms of natural selection. According to [7], the main features of these algorithms include: dispensing information about the function gradient, the functions involved do not have to be continuous and, when well configured, may avoid local minima.
[8] used the augmented simulated annealing method for optimization of reinforced concrete beams. The main objective of the optimization was to minimize the structure's cost, taking into account resistance and service parameters. Among the optimization variables, the dimensions of the cross-sectional areas of the beam are available. The steel area required in each span (upper and lower) and the numbers of bars in each section were also considered as decision variables. Similarly, the steel area required for the transverse reinforcement (at three span's regions) and the required spacing were also considered. [9] also studied the use of the augmented simulated annealing algorithm in the optimization of reinforced concrete structures, in particular containment structures.

Simple genetic algorithms implemented in FORTRAN were employed by [10] for the optimized detailing of continuous reinforced concrete beams. Unlike other works, only the cross-sections of the beams are considered as decision variables. The reinforcement and their detailing are calculated after the cross-sections are defined for each iteration. The detailing was carried out in two stages, based on pre-defined configurations for the reinforcements. The objective function was taken as the cost of the structure, composed of the cost with concrete, steel and shapes. The restrictions were bound to the minimum width of the beams to support the bars, limit ratio between beam width and height, maximum moment check, neutral line height verification, minimum reinforcement amount reaching the support, maximum and minimum reinforcement rate, and boundary dimensions for the cross-sections.

The isolated optimization of reinforced concrete pillars was studied by [11]. In this work, columns subjected to multiple loading conditions are designed and optimized using mathematical optimization techniques for non-linear problems and a graphical method.

[12]have implemented genetic algorithms for the design and optimization of planar reinforced concrete frames. The total structure's cost was considered as an objective function, having as variables the dimensions of the cross-sections, diameter, number of bars and reinforcement topology. Among the constraints there are displacement limitations, number of bars per reinforced layer, boundary dimensions of sections, bending and shear strength, and structure displacements. Similar work was also performed by [13], [14] and [15].

[7] also studied the optimized sizing of reinforced concrete planar frames using genetic algorithms. For the optimization, a database of candidates for typical sections of reinforced concrete was created, containing pre-established reinforcement configurations, which already satisfy normative limitations. These configurations differ in section size, rate, and reinforcement layout. For optimization, a simple genetic algorithm was employed and the objective function was the structure's cost. A methodology similar to [7] was used by [16].
Other types of meta-heuristic algorithms were used by [17] (Big Bang -Big Crunch algorithm) and [18] (Harmony Search Method algorithm), for the optimization of reinforced concrete frames.

The optimization of the cost and the arrangement of the elements for reinforced concrete space frames was the objective of the research of [19] . In his studies, the Ant System algorithm was employed to search for the lowest cost of the structure by changing the dimensions of its cross-sections and the arrangement of its columns. It is important to note that the methodology presented by [19], with regard to optimization of the column arrangement, simply seeks the optimal dimensions of the span between columns in two orthogonal directions, not being obvious the possibility of disposition in arbitrary directions. Although this methodology can be efficient for structures with rectangular shapes and where the frames are regularly spaced, it may not display the same efficiency for irregular topologies.

III. FORMULATION OF OPTIMIZATION PROBLEM

For the current projects of reinforced concrete buildings, the positioning of the beams in a floor, mostly, is conditioned by the masonry positions. Although this consideration is not necessarily restrictive, it will be adopted in this work for simplification purposes that the arrangement of the beams is determined prior to the optimization process. It is important to note that the assumption that the beams are bound to the masonry position is just a simplifying consideration and there is no need for the masonry existence, as the example shown in Fig. 1.

Figure 1. Example of the predefinition of the beams of a floor.

Similar to beams, slabs should be predefined by the user, due to the definition of their panels being delimited by the beams. Thus, the slabs will remain unchanged during the optimization process, although they are considered in the analysis of the model.

3.1. DIMENSIONS OF CROSS-SECTIONS

Although the arrangement of the beams remains unchanged during the optimization process, the dimensions of their cross-sections can be altered at each iteration. To do so, it is necessary to define the maximum and minimum limits allowed for the dimensions of the cross-sections to be delivered during the optimization process. Due to the smaller influence of the cross-section widths in the bending strength of the beams, the section width is considered constant and equal for all the beams. Thus, only the maximum and minimum limits for the height of the cross-sections should be defined, respectively, $h_{max}^{(b)}$ and $h_{min}^{(b)}$.

Due to the ease and economic construction, it is common to limit the number of distinct sections used in a project. This consideration provides savings in shape-making, greater ease of execution and labor savings. However, the maximum number of distinct sections (s_{beams}) to be considered in the project is the designer's choice and there is no consensus or general rule that limits its use. Therefore, in this work, the maximum number of distinct sections allowed during the optimization process is defined by the user.

A similar strategy will be used for the columns, where the maximum number of distinct sections for the columns ($s_{columns}$) shall be arbitrated by the user. Similar to the beams, the smallest dimension of the column is constant for all the columns, with the largest size being variable, and its upper limits are $h_{max}^{(c)}$ and $h_{min}^{(c)}$, respectively.
The conditions herein defined for section dimensions can be summarized by equations (1) and (2) respectively for the sections of beams and columns.

$$h_{min}^{(b)} \leq h_i^{(b)} \leq h_{max}^{(b)} \quad (1)$$

$$h_{min}^{(c)} \leq h_i^{(c)} \leq h_{max}^{(c)} \quad (2)$$

In addition to the maximum amount of sections used for columns and their limit dimensions, it is intended, in the next work stages, to consider variable the number of columns in the structure, participating in the optimization process and not having a fixed value.

3.2. COLUMN POSITIONING

The strategy for the column positioning of the structure is one of the most complex stages for modeling this optimization problem. It will be considered that the positioning of the columns will be restricted to the domain of the beams, that is, it will only be possible to position columns that intercept beams, the positioning of columns in direct contact with the slabs will not be allowed.

In the literature, the column positioning, when considered, is parameterized through the construction of the uniform distribution in perpendicular directions, similar to matrices, as considered by [19] . In another strategy, less popular, the positioning is given by the coordinates of the geometric center of the section of the column

on the floor plan, then it is verified if the interception with beams occurs.
The procedure for positioning the columns in this work consists of: 1) transforming each floor into a parameterized equivalent beam, 2) defining the positioning of the column in this equivalent beam, 3) identifying in which beam element of the floor and 4), in which position the column should be placed. The stages of the procedure above are schematically represented in Fig. 2.

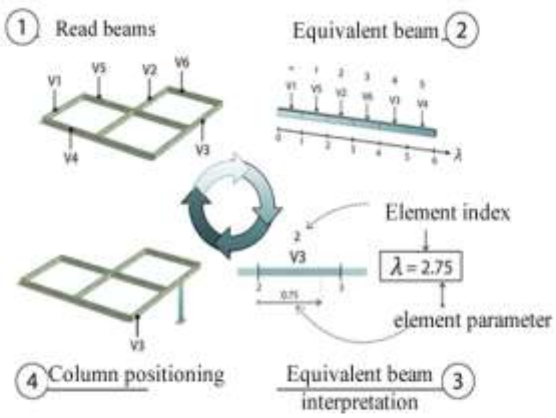

Figure 2. Schematic procedure to column positioning.

In stage 1 of Fig. 2, the floor beams are identified and stored in an ordered list, being possible to access them later through an index, j, corresponding to its position in the list, where $0 \leq j < n_{pav}$, and n_{pav} is the amount of beams on a floor.

In stage 2, the floor beams are transformed into an equivalent beam. The equivalent beam is composed of n_{pav} beams with unitary length, positioned in series, respecting the order of the elements in the list of floor beams created in stage 1. It is important to note that the equivalent beam will have the total length equal to the number of beams present in the floor.

Due to the adoption of the strategy of converting a floor into a single equivalent beam, the positioning of a column can be defined through a single positioning parameter, λ_i, which will identify in which position of the equivalent beam the column should be positioned. This strategy allows reducing the number of variables needed to define the positioning of the columns, being one of the additional contributions of this work. It is important to highlight that among the strategies present in the literature, two variables are necessary to characterize column positioning. In a strategy, the coordinates of the columns are considered in order to define their position, and two variables would be necessary (for the coordinates x_i and y_i of the column), besides the verification of the intersection with the beams. In another strategy, the positioning of a column is characterized by a variable in order to identify in which element the column is to be positioned, and a second variable that identifies in which position this element should be positioned. For example, suppose you want to position 15 (fifteen) columns in a structure. In the strategies adopted in the literature, two variables would be necessary for each column, that is, 30 (thirty) variables to represent the positioning of the 15 (fifteen) columns. In the strategy adopted in this work, it will be necessary to define only one variable per column, that is, 15 (fifteen) variables, reducing by half the number of variables required for the columns and, consequently, the complexity of the problem.

From the positioning parameter, λ_i, it is possible to define in which position of the equivalent beam the column should be allocated, as shown in stage 3 of Fig. 2. Because the positioning parameter is a decision variable, its permissible limits must be defined, according to equation (3):

$$0 \leq \lambda_i < (n_{pav}) \quad i = 1, 2, \ldots, n_{columns} \tag{3}$$

If λ_i respect the equation (3), the index in the list of floor beams of the element in which the column must be placed will be given by the integer part of λ_i, while its relative position in this element is given by the decimal part of λ_i and $n_{columns}$ is the number of columns considered. Therefore, the index in the list of floor beams of the element in which the column is to be positioned is given by equation (4), and the relative position of the positioning in this column is expressed by the equation (5):

$$j = \lfloor \lambda_i \rfloor = \mathbf{max}\{m \in \mathbb{Z} \mid m \leq \lambda_i\} \tag{4}$$

$$\bar{\lambda}_i = \lambda_i - j \tag{5}$$

where j is the index in the list of floor beams of the element where the column should be positioned, $\lfloor \lambda_i \rfloor$ is the floor function (or rounding down) of the positioning parameter in the equivalent beam, and $\bar{\lambda}_i$ is the relative positioning parameter of the column in the j-th beam of the list of floor beams. Fig. 3 illustrates the use of the $\bar{\lambda}_i$ parameter for the positioning of a column in the j-th floor beam.

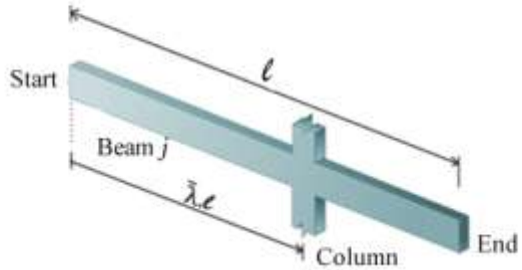

Figure 3. Column parameter in a beam.

3.3. DESIGN VECTOR

In an optimization project, the decision variables of the problem must be assembled into a single vector, named design vector. The solution vector will specify a solution in the optimization process, modified with each interaction. Since the design vector contains each of the decision variables, the amount of information contained in it will vary for each problem analyzed.

For the problems herein formulated, the solution vector must contain the following decision variables: the heights of the sections of the beams ($\mathbf{x}^{(1)}$), which section should be allocated to each beam ($\mathbf{x}^{(2)}$) the maximum dimensions of the sections of the columns ($\mathbf{x}^{(3)}$), which section should be allocated to each column ($\mathbf{x}^{(4)}$) and the position parameter λ_i for each column ($\mathbf{x}^{(5)}$). The composition of the design vector $\mathbf{x}$ is shown schematically in Fig. 4.

$$\mathbf{x}^{\mathrm{T}} = \begin{Bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \mathbf{x}^{(3)} \\ \mathbf{x}^{(4)} \\ \mathbf{x}^{(5)} \end{Bmatrix}^{\mathrm{T}} \begin{matrix} \longrightarrow \text{section dimensions of beams} \\ \longrightarrow \text{beam's section id} \\ \longrightarrow \text{section dimensions of columns} \\ \longrightarrow \text{column's section id} \\ \longrightarrow \text{column position} \end{matrix}$$

Figure 4. Schematic representation of the design vector.

A more formal definition, the design vector $\mathbf{x}$, is presented in equation (6), its components being defined by the following equations (7) to (11):

$$\mathbf{x} = \{\mathbf{x}^{(1)} \quad \mathbf{x}^{(2)} \quad \mathbf{x}^{(3)} \quad \mathbf{x}^{(4)} \quad \mathbf{x}^{(5)}\} \qquad (6)$$

$$\boldsymbol{x}^{(1)} = x_i^{(1)} \qquad i = 1, 2, \ldots, s_{beams} \qquad (7)$$

$$\mathbf{x}^{(2)} = x_k^{(2)} \qquad k = 1, 2, \ldots, n_{beams} \qquad (8)$$

$$\mathbf{x}^{(3)} = x_m^{(3)} \qquad m = 1, 2, \ldots, s_{columns} \qquad (9)$$

$$\mathbf{x}^{(4)} = x_p^{(4)} \qquad p = 1, 2, \ldots, n_{columns} \qquad (10)$$

$$\mathbf{x}^{(5)} = x_p^{(5)} \qquad p = 1, 2, \ldots, n_{columns} \qquad (11)$$

$n_{columns}$ and n_{beams} are the quantity of columns and beams of the model, respectively. It is important to emphasize that the results presented in this study will neither contemplate the variation of column positioning nor the variation of its section because these functions are in ongoing development.

3.4. OBJECTIVE FUNCTIONS

In multiobjective optimization problems,theobjective function is represented by a vector of objective functions. Up to the current version of this work, the total cost and the maximum displacement of the structure are adopted as objective functions.

The total cost of the structure considered in this work is related to the consumption of materials, and the labor cost will be taken into account indirectly and approximately through the unit costs of the materials. The cost function is defined according to equation (12):

$$f_1 = \sum_{i=1}^{N} C_c \cdot V_{c_i} + C_s \cdot \gamma_s \cdot A_{s_i} \cdot L_i + C_f \cdot A_{f_i} \qquad (12)$$

where C_c, C_s and C_f are the unit cost per concrete volume, per steel weight and per shape area, respectively; N is the amount of elements in the model, V_{c_i} is the volume of concrete, A_{s_i} is the steel area, L_i its length and A_{f_i} the shape area of the $\boldsymbol{i}$-th element of the structure.

The global maximum displacement function consists in finding the maximum displacement in the structure, in absolute value. The maximum displacement is obtained by navigating through elements of the model and evaluating if these displacements are the largest in absolute value in the structure. The maximum displacement function is formalized in equation (13):

$$f_2 = \max(\boldsymbol{\delta}) \qquad (13)$$

where $\boldsymbol{\delta}$ is the displacement vector of the structure.

It is intuitive to observe that increasing sections in structures subjected to flexing tends to increase their rigidity and, consequently, reduce displacements. However, the increase of the sections will cause a growth in the consumption of materials, raising the cost with materials of the structure. Therefore, it can be noticed that the two functions tend to present an inverse correlation, that is, if one value increases, the other tends to decrease. Thus, in order to minimize both objective functions simultaneously, one must not improve in detriment of the other, seeking the Pareto Frontier for the problem, similar to the one presented in Fig. 5.

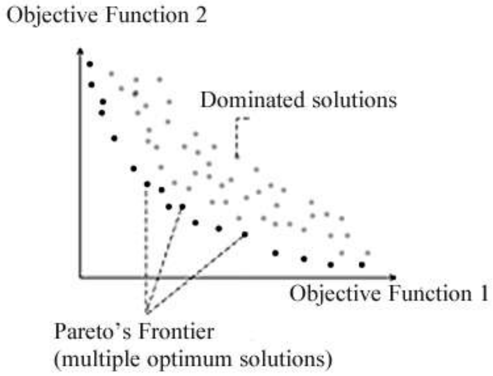

Figure 5. Example of a Pareto Frontier.

3.5. CONSTRAINTS

In this paper, the restrictions imposed for the optimization problem refer to the maximum and minimum dimensions of the cross-sections of the elements and maximum displacements. Differently from most of the works presented in the literature, the resistant capacity of the elements is not directly verified, the elements are dimensioned to resist the requesting efforts and, if sizing is not possible, the objective functions are penalized.

Due to the integration with the Robot, the verification of limiting displacements, the design and eventual verification of the resistant capacity of the elements, as well as the verification of crack openings, can be verified by the Robot itself, depending on the adopted standards for the project and customizations of the designer. If there is a violation of the limits configured in the Robot itself, calculation alerts are issued by the robot, informing the project about the nonconformity in regards to the defined parameters. It is anticipated that, at the current stage of implementation, any alerts of tampering with the sizing and verification conditions issued by the Robot, irrespective of their nature, are treated simply as a violation of project conditions, penalizing the objective functions.

The penalties for the objective functions amplify the calculated value of the functions by weighting, so that the project that presents restraint violations can be disadvantaged in the optimization process, decreasing its chances of being present in the final optimal solution. The penalty function adopted in the present study is similar to that adopted by [20] , modified to a generalized magnitude, and is displayed in equation (14):

$$P_t(\boldsymbol{x}) = h \cdot \left[\sum_{k=1}^{N} \left| \frac{\mu_k(\mathbf{x}) - \mu_k^{max}}{\mu_k^{max}} \right| \right] \qquad (14)$$

where N is the number of elements in the structure, h is the weight of the penalty, commonly defined as a high number, x is the analyzed design vector, μ_k is the value of a certain physical quantity that one wishes to evaluate and μ_k^{max} is the maximum value allowed for this very quantity.

IV. IMPLEMENTATION

The structure analysis will be performed using the Autodesk Robot Structural Analysis ® 2017 (Robot). The Robot is a finite element structural analysis program with BIM (based information modeling) technology maintained by Autodesk. With Robot, it is possible to analyze linear analysis of bar structures until complex finite element models, with computational fluid simulation for dynamic analysis, considering effects caused by wind (see Fig.6).

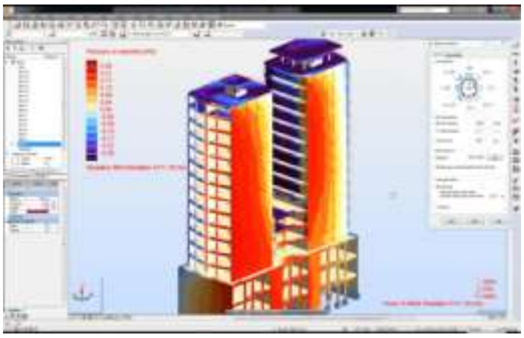

Figure 6. Program interface of the Autodesk Robot Structural Analysis 2017.

In the current work, Robot is used as an environment tool for the modeling and configuration of the initial structure, processing and visualization of the final results.

It is via graphical input that the user must define the position of the beams and the initial dimensions of its sections, the slab panels and the columns that the user wishes to preserve during the analysis. The change on the sections of the elements will happen from the modeled structure by the addition and modifications in the position of the new columns, based on the number of columns desired by the user.

It is important to emphasize that some care must be taken when defining the structure model so that the methodology employed here works correctly. Among those, lies the definition of the element type for beams and columns. In Robot, linear elements can be chosen between a beam, column or bar. For the sizing and positioning to occur as expected, the linear elements representing the beams must be defined in the Robot by choosing the option "Beams" within the attribute "Structural Object", being also inserted as CA Beams (can be verified in the “Type” attribute), in reference to reinforced concrete beams.

Evolutionary algorithms are algorithms that mimic the evolutionary behavior of individuals. In the nomenclature of evolutionary algorithms, each solution is called an "individual". The set of individuals generated in an algorithm iteration (also called generation), is named "population". During each iteration, dominant individuals, with greater aptitude to improve the solution, named "parents", are selected and are congregated through the combination and mutation procedure to generate new individuals, that is, new solutions. When a part of the population is kept unchanged between iterations, "elitism" is said to occur. Elitism is a strategy adopted to preserve dominant individuals, those who have good fitness across generations against mutation and crossing procedures. At each generation, the following procedure is repeated until the end of generations is reached or an convergence criterion is established: a random population of individuals is generated, the objective function is evaluated for each individual of the population, and a new population must be created; the creation of a new population occurs through the selection of the parents, crossing between the parents with or without mutation (slight alteration in the characteristics inherited from the parents); then the previous population is replaced by the new population, with or without elitism, and the process is repeated until a stopping criterion is reached.

NSGA-II (Non-dominated Sorting Genetic Algorithm II) is an elitist genetic algorithm developed by [21] , for the solution of multi-objective optimization problems. The main feature of NSGA-II is the non-dominant rapid classification procedure, where a population is classified in borders formed by non-dominant individuals , that is, everyone has equal chances of reproduction. At each iteration of this algorithm, individuals are identified in order to form the Pareto Frontiers, based on the concept of non-dominance [22], following the standard procedures of the evolutionary algorithms. Another important feature of NSGA-II is the preservation of the diversity of solutions [21] , that is, individuals tend to be well distributed over Pareto Frontiers within each generation.

The NSGA-II algorithm presented in this work is available in the jMetal library, elaborated by [23], originally for Java language, also having a C# version, the one used in this work. For more details on NSGA- II and its implementation, the reader should check the works of [23] and [21] since its direct implementation is not one of the objectives of this work.

4.1. ROBOT-NSGA-II INTEROPERABILITY

In order to optimize the structure defined in the Robot, it is necessary to integrate it with a NSGA-II optimization algorithm. The integration can be performed through the computational implementation of a program that uses the Robot API and integrate with the jMetal library, which includes NSGA-II among its algorithms.

Therefore, a computational program that allows integration between the Robot API and NSGA-II is being developed in this work. Until the present version of this work, the program consists of six main modules: ControlPanel, Flx, RobotCon, Model, Optimization and NSGA-II (see Fig. 7).

The ControlPanel module is responsible for the user-friendly graphical interface, data entry, and presentation of results. The Flx is responsible for the communication between the interface and other modules. It is in this module that the interface data is read and transmitted to the other modules. The Model is responsible for storing the values of the decision variables, and where the determination of the constraints and objective functions are evaluated. It is important to highlight that it is in the Model module that the Optimization module performs the optimization process of the structure, defining the optimization model and optimization parameters, such as population size, generations, mutation and combination parameters. Within the Optimization module, the NSGA-II is executed, in which the operations of the optimization process are executed effectively. However, after performing the optimization, the results and parameters of the analysis are stored in the Optimization module, being transferred successively to the modules Flx and ControlPanel, where they are presented to the user. The integration and relationship between modules are presented Fig. 7.

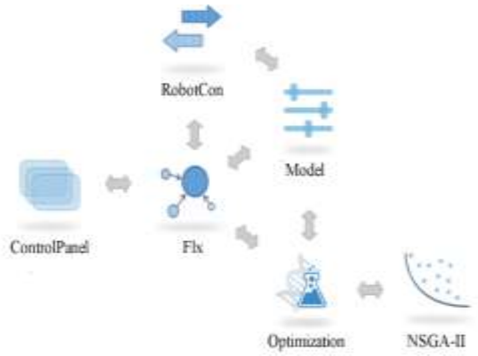

Figure 7. Interaction between modules.

V. RESULTS

To illustrate the application of the proposed methodology, examples and some results obtained will be presented and commented below. In all cases, the floor geometry, through the positioning of the beams and slabs, was pre-determined in the Robot's graphical interface using its native functions, as well as the types, cases and combinations of structure loadings, type and material properties. It is important to note that in all the examples, it is not necessary to define an initial positioning for the columns, being those, as well as all the others, defined by the optimization process, requiring no previous information on the positioning of the columns.

It should also be highlighted that in the examples presented, it is considered that there is no limitation for column positioning. Although not contemplated in the present version of this work, the authors already have tests where there are considered restrictions on the positioning of the columns, for example, in function of the positioning of doors and windows.

5.1. EXAMPLE 1

The examples 1 to 4 seek to explore the possibilities of the proposed methodology, presenting 4 cases with different approaches. In all cases, a 3-storey building will be used, with the floor geometry shown in Fig. 8. The distance adopted between the floors was 3 m.

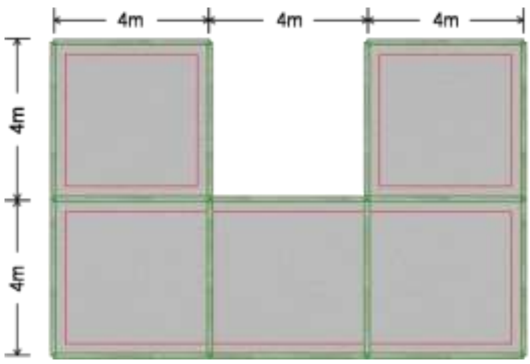

Figure 8. Floor plan for Example 01 to 04.

It was considered that a uniformly distributed load of 5.85 $kN.m^{-1}$, equivalent to a 14 cm thick masonry and 3 m high in stoneware ceramic brick, will be applied to each of the beams. In addition, a uniformly distributed load of 3 $kN.m^{-2}$ was applied to the slab panels. The weight of the elements was also considered. Twenty individuals per generation in a total of 100 generations were used and the standard NSGA-II configurations provided by the jMetal library [23].

As used in the work of [6] 100 individuals were used (except in the first iteration, where 200 individuals are analyzed to increase the initial diversity) and 100 generations in each case. The NSGA-II parameters used were the predefined ones, namely: mutation rate of 0.50; mutation probability of 0.10; crossover rate of 0.80; and elitism proportion of 0.50, i.e., maintenance of half the population i in the iteration i+1.

In this case, only one section was allowed for all the beams of the building. The width was fixed at 14 cm and its height varied along the iterations between 30 and 70 cm. Similarly, only one section was considered for all columns. The smallest cross-sectional dimension was adopted as 14 cm. The largest dimension was considered variable between 30 and 50 cm. The objective functions chosen were the cost of materials (concrete, steel and shapes) versus the maximum vertical displacement on the floor. The only constraint imposed was that the maximum displacement in each beam did not exceed the ratio between its length and 350. The displacement limitation can be configured in the Robot itself, which, when violated, emits a sizing error and the project that generated it gets penalized.

With the parameters mentioned above, the set of solutions presented in Fig. 9 was obtained after 100 iterations, in which the graph shows the Pareto Frontier considering the displacements in centimeters and the cost in thousands of Brazilian Real (R$).

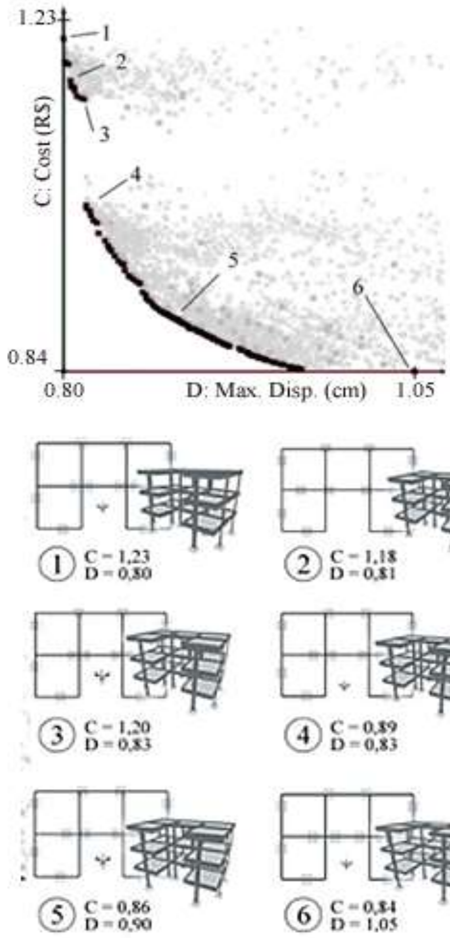

Figure 9. Solution space for Example 1.

By observing Fig. 9, one can perceive a discontinuity in the Pareto Front for this case. After inspection of the individuals in solutions 3 and 4 highlighted in the figure, it was verified that, although their topologies are similar, the difference lies in the height of the beams for each case. Individual nº 3 has 61 cm high beams, while individual nº 4 is 60 cm high. This small variation in beam height causes a large difference between the costs of the two groups. This occurs because beams with a height greater than 60 cm require the use of skin reinforcement, causing a considerable increase in reinforcements, evidenced by the discontinuity of the Pareto Frontier obtained.

In addition, it can be observed that in the lower part of the Pareto Front, regarding the solutions with beams with less than 60 cm, that small cost variations (resulting in economy) causes a proportionally greater increase in the displacements, while the upper part presents a practically proportional variation between maximum displacements and cost.

Regarding column positioning, it is observed that the difference between the highlighted solutions is practically has no diversity, a fact that was also observed in the other solutions that compose the Pareto Frontier of Fig. 9.

5.2. EXAMPLE 2

In order to verify if the height variation of the beams and columns influenced the lack of diversity of the solutions in Example 1 regarding column positioning, Example 2 presents the same problem as Example 1, considering, however, that the sections of the elements are pre-established and non-variant throughout the optimization process. Therefore, beam height was kept equal to 50 cm and the largest dimension of the columns in 40 cm. Thus, the parts of the cost pertaining to the concrete and shapes will be fixed, being able to only vary the cost with armature. Fig. 10 presents the results for this case.

Based on Fig. 10, it can be seen that the fixation of the dimensions of beams and columns did not produce significant changes in the topology of the structure, presenting a column positioning very similar to those of Example 1 (see Fig. 9).

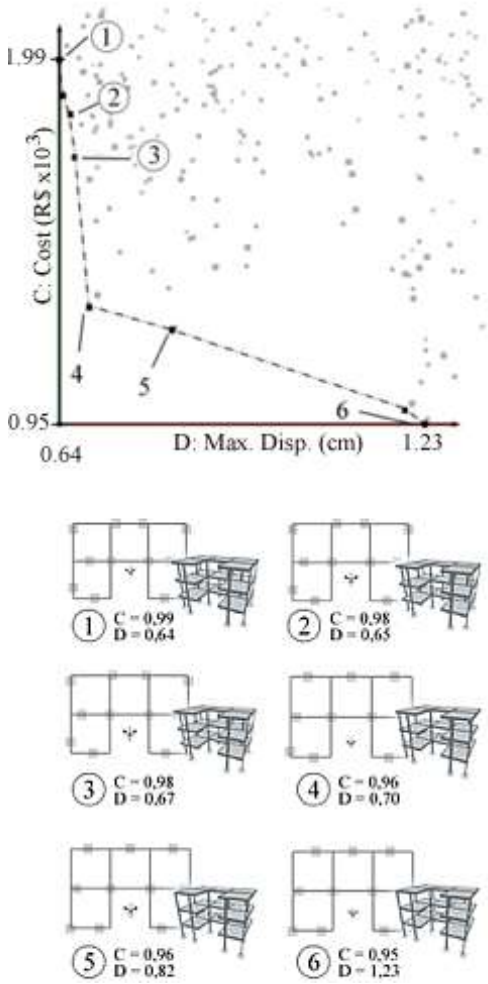

Figure 10. Solution space for Example 2.

5.3. EXAMPLE 3

From the experiment with Example 2, Example 3 presents the consideration of the absolute maximum moment as one of the objective functions instead of the structure's cost. Therefore, the optimization is performed in order to minimize the absolute bending moment in the beams, along with the maximum displacement. Since the longitudinal reinforcement is a function of the bending moments of the beams, it is believed that minimizing the maximum absolute bending moment of the structure indirectly limits the maximum steel area required in the beams. The solutions obtained are shown in Fig. 11.

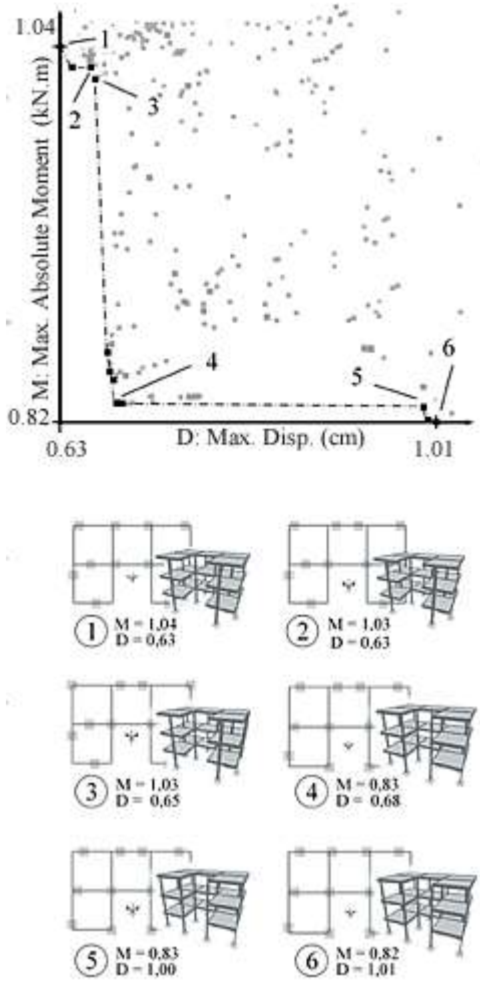

Figure 11. Solution space for Example 3.

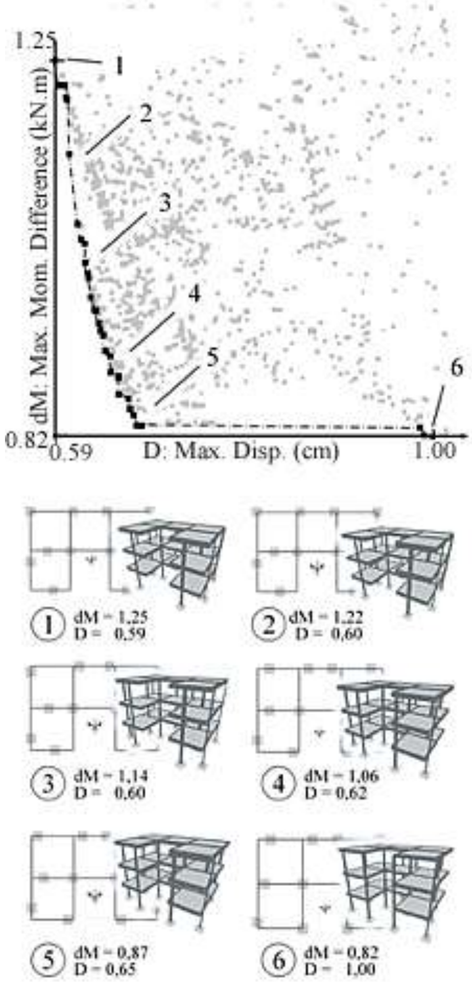

Figure 12. Solution space for Example 4.

5.4. EXAMPLE 4

Here, it was sought to use a criterion similar to that used by [24]. In addition to the maximum displacement in the beams, the difference between the maximum positive and negative moments was adopted as objective function in order to obtain uniformity in the values of the bending moments in the structure, varying the sections of the beams, which have a direct influence on the displacements. The results for this case are shown in Fig. 12.

By observing Fig. 12, it is possible to conclude that large variations on the difference between the moments result in discrete variations in the displacements, with the exception of the individual nº 6. Individual nº 6 presents the smallest section among those presented in Fig. 12, which certainly influenced the discrepancy of its displacement in relation to the other individuals in the solution.

It is important to point out that in this case, the solutions differed more between one another with respect to the positioning of the columns, being the case 6 a possible solution, close to what would be proposed by a designer in a daily project.

5.5. EXAMPLE 5

Example 5 intends to show the versatility of the proposed methodology regarding the floor shapes. In the literature, as shown in Section II, the methodologies for the optimum positioning of columns are only used in regular topology structures, often with floors with regular spaced beams and perpendicular to each other. From this example, it is intended to show that the methodology is able to handle floors with non-regular forms such as, for example, non-orthogonal elements and even curved beams.

The shape of the structure's floor is shown in Fig. 13, with the spacing between floors equal to 3 m. A load uniformly distributed across all slabs equal to 3.75 kN.m^{-2} and a uniformly distributed load of 5.85 kN.m^{-1} on the beams, referring to the masonry, was considered, according to Example 1.

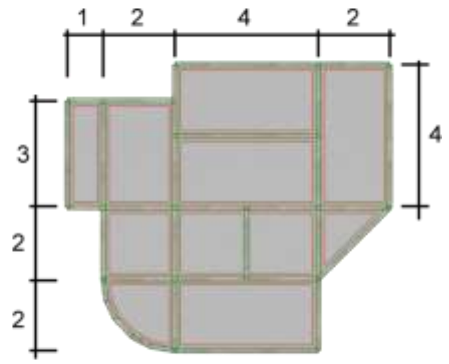

Figure 13. Floor Plan for Example 5.

Similar to Example 4, the objective functions adopted were the difference between the maximum positive and negative momentum, and maximum beam displacement. 16 columns were chosen at launch. In addition, the height of the beams was considered variable along the iterations, with intervals similar to those of Example 1.

It was decided to change the parameters of the algorithm in order to evaluate its influence on the results. Thus, for Example 5, a mutation probability of 0.20 (20%) was considered, unlike the 0.10 (10%) adopted in all examples. The results for this case are shown in Fig .14.

It is observed in Fig. 14 that the change in the mutation parameters did not provide an increase in the diversity of the results, especially concerning column positioning, producing very similar results among each other. However, it is considered that the solutions obtained by the methodology yielded feasible results with small practical changes, as long as there is no limitation on the positioning of the columns.

VI. CONCLUSION

The methodology suggested in this work proved to be a tool with potential for exploring the solution space of the problem, capable of finding a set of optimal solutions to a problem proposed by altering the sections of the elements and the positioning of the columns. Although the results presented may be questionable concerning their feasibility and compliance with practical recommendations, mainly due to the preferences of the designer and regional practices, for the most part, regardless of the objective functions adopted, the results have produced solutions that can serve as an aid in the structural conception.

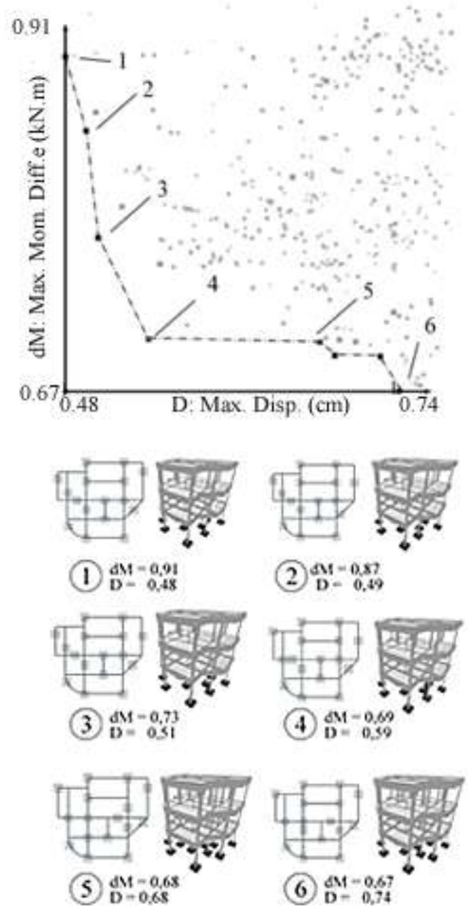

Figure 14. Solution space for Example 5.

As can be seen in the comments made on the examples, the main objective has never been to find optimal structures, although this is the main function of the algorithm, but rather to employ optimization techniques in order to produce structural suggestions as an aid to the designer. However, it is inevitable to realize that the proposed methodology needs to be improved, especially with regard to increasing the genetic diversity of the solutions. In all the examples presented, one can note the presence of very similar solutions, especially concerning column positioning. This may be related to the way the problems were formulated, for example, in the objective functions adopted and in the approximate design criteria.

The present work also brings as contribution an alternative methodology to the positioning of the columns, as demonstrated in item 3.1.3, which is able to reduce considerably the number of variables involved in the problems. In addition, the search for the positioning of the columns had previously been presented only for regular topology structures on the floors, mostly in rectangular and perpendicular shapes. Another interesting feature of the work is the integration between a commercial program aimed at the design of structures with a multi-objective optimization algorithm, which is still a very little explored alternative.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by the Federal Institute of Ceara.

REFERENCES

[1]. R. Hicks, P. A. Henne, Wing Design by Numerical Optimization, Journal of Aircraft, 15(7), 1978, 407-412.
[2]. D. A. Swanson, H.T. Wu, H. Ashrafiuon, Optimization of aircraft engine suspension systems, Journal of Aircraft, 30(6), 1993, 979-984.
[3]. S.A. Mirhassani, A. Yarahmadi, Wind farm layout optimization under uncertainty, Renewable Energy, 107, 2017, 288-297.
[4]. G. Xiao, Z. Juan, C. Zhang, Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization. Transportation Research Part C: Emerging Technologies, 71, 2016, 447-463.
[5]. K.C., Sarma, H. Adeli, Cost optimization of concrete structures, Journal of Structural Engineering, 124(5), 1998, 570-578.
[6]. M. Babaei, M. Mollayi, Multi-objective optimization of reinforced concrete frames using NSGA-II algorithm, Engineering Structures and Technologies, 8(4), 2016, 157-164.
[7]. C. Lee, J. Ahn, Flexural design of reinforced concrete frames by genetic algorithm, Journal of Structural Engineering, 129(6), 2003, 762-774.
[8]. M. Leps, M. Sejnoha, New approach to optimization of reinforced concrete beams, Computers and Structures, 81, 2003, 1957-1966.
[9]. F.G. Vidosa, V. Yepes, J. Alcalá, M. Carrera, C. Perea, I. Paya-Zaforteza, Optimization of reinforced cocnrete structures by simulated annealing, in C. M. Tan (Ed), Simulated Annealing.(Vienna, Austria, 2008), 307-320.
[10]. V. Govindarah, J.V. Ramasamy, Optimum detailed design of reinforced concrete continuous beams using genetic algorithms, Computers and Structures, 84, 2005, 34-48.
[11]. H. J. Lee, Optimum RC column reinforcement considering multiple load combinations, Structural Multidisciplinary Optimization, 39, 2009, 153-170.
[12]. C. V. Camp, S. Pezeshk, H. Hansson, Flexural design of reinforced concrete frames using a genetic algorithm, Journal of Structural Engineering, 129(1), 2003, 105-115.
[13]. M. J. Fadaee, M. H. Grierson, Design optimization of 3D reinforced concrete structures, Structural Optimization, 12, 1996, 127-134.
[14]. A. Guerra, P.D. Kiousis, Design optimization of reinforced concrete structures, Computers and Concrete, 3(5), 2006, 313-334.
[15]. A.A.A. Aga, F. M. Adam, Design optimization of reinforced concrete frames, Open Journal of Civil Engineering, 5, 2015, 74-83.
[16]. H.G. Kawak, J. Kim, Optimum design of reinforced concrete plane frames based on predetermined section database, Computer-Aided Design, 40, 2008, 396-408.
[17]. A. Kaveh, O. Sabzi, Optimal design of reinforced concrete frames using big band-big crunch algorithm, International Journal of Civil Engineering, 10(3), 2012, 189-200.
[18]. M. Kripka, Optimization of reinforced concrte frames by Harmony Search Method, Congress on structural and multidisciplinary optimization, Sydney, 2015.
[19]. P. Sharafi, M.N.S. Hadi, L. Teh, Heuristic approach for optimum cost and layout design of 3D reinforced concrete frames, Journal of Structural Enginnering, 138(7), 2012, 853-863.
[20]. L.F.F. Miguel, L. H. Lopez, L.F.F. Miguel, Multimodal size, shape and topology optimization of truss structures using the Firefly Algorithm, Advances in Engineering Software, 56, 2012, 23-37.
[21]. K. Debs, A. Pratap, S. Agarwal, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, 6(2), 2002, 182-197.
[22]. A. H. Gandomi, X. S. Yang, S. Talatahari, Metaheuristics applications in structures and infrastructures, (Waltham: Elsevier, 2013).
[23]. J. J. Durilo, A. J. Nebro, jMetal: A Java framework for multi-objetive optimization. Advances in Engineering Software, 42(10), 2011, 760-771.
[24]. M. Kripka, Determinação do posicionamento ótimo dos apoios em edificações analisadas pelo modelo de grelha, doctoral diss. (in portuguese), University of São Paulo, São Carlos, SP, 1998.